

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2002-259186**

(43)Date of publication of application : **13.09.2002**

(51)Int.Cl.

G06F 12/00

(21)Application number : 2001-061081

(71)Applicant : **HITACHI LTD**

HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 06.03.2001

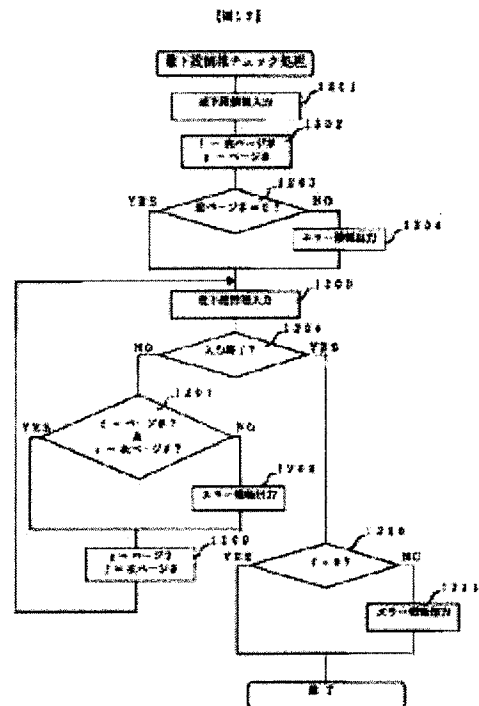
(72)Inventor : OTAWARA MINORU

(54) METHOD, PROGRAM AND DEVICE FOR CHECKING AND PROCESSING COMPATIBILITY OF TREE STRUCTURED INDEX

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method, a device and a program for checking and processing index compatibility of a database to enable compatibility check of a database even when problems such as destruction and a defect are generated in the index itself and to specify/detect a destructed place of data structure.

SOLUTION: When pieces of information at the respective stages are created, check with the information at an upper page and in page check afterward, control is performed so that unauthorized page information is not used.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-259186

(P2002-259186A)

(43) 公開日 平成14年9月13日 (2002.9.13)

(51) Int.Cl.⁷

G 0 6 F 12/00

識別記号

5 3 1

5 2 0

F I

G 0 6 F 12/00

テーマコード* (参考)

5 3 1 R 5 B 0 8 2

5 2 0 A

審査請求 未請求 請求項の数 3 O L (全 10 頁)

(21) 出願番号 特願2001-61081(P2001-61081)

(22) 出願日 平成13年3月6日(2001.3.6)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 大田原 実

神奈川県横浜市中区尾上町6丁目81番地

日立ソフトウェアエンジニアリング株式会社
社内

(74) 代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5B082 DC02 DC05

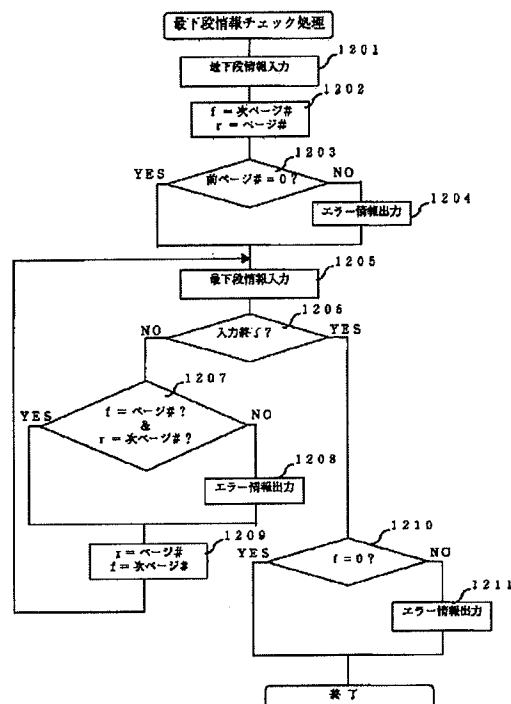
(54) 【発明の名称】 TREE構造型インデクスの整合性チェック処理方法およびプログラムおよび装置

(57) 【要約】

【課題】 インデクス自体に破壊や欠損などの問題が生じている場合であってもデータベースの整合性チェックを可能とし、データ構造の破壊個所を特定・検出するデータベースのインデクス整合性チェック処理方法および装置およびプログラムを提供することにある。

【解決手段】 各段の情報作成時に、上段ページ情報とのチェックを行い、以降のページチェックでは、不正ページ情報が使用されないように制御する。

【図12】



【特許請求の範囲】

【請求項 1】 データベースに接続したインデックス整合性チェック処理装置において、

前記インデックス整合性チェック処理装置は、前記データベースに格納されたデータをデータが格納された順に前記データの入力を受け付ける手段と、

前記データベースに格納されたデータのデータ構造情報を取得する手段と、前記データ構造情報に基づいてデータを最上段から最下段まで段毎に並び替えを行う手段と、前記最下段の並び替え結果をもとに、データ間のポインタ情報の整合性をチェックする手段とを備えたことを特徴とするインデックス整合性チェック処理装置。

【請求項 2】 データベースに格納されたデータ構造の整合性のチェック方法において、

前記データベースに格納された順序に従ってデータの入力を受け付け、

前記受け付けたデータ間のポインタ構造情報を格納し、前記構造情報に基づいて階層ごとにデータの並べ替えを行い、前記データの並べ替え結果に基づいてデータ間のポインタ構造の整合性の判定を行うことを特徴とするデータベースに格納されたデータ構造の整合性のチェック方法。

【請求項 3】 TREE 構造のデータが格納されたデータベースと計算機とが接続されたシステムにおけるインデックス整合性チェックプログラムにおいて、

前記プログラムは、前記データベースに格納された TREE 構造のデータの格納情報を取得し、

前記データの格納情報に基づいて、前記 TREE 構造の階層ごとにデータを並べ替え結果と、前記 TREE 構造の最下位の階層でのデータの並べ替え結果とを比較し、前記比較結果に基づいて、前記データベースに格納された TREE 構造のデータのデータ構造の整合性のチェックを行うことを特徴とするインデックス整合性チェックプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、TREE 構造型インデックスを備える DBMS（データベース管理システム）におけるインデックス整合性のチェックに関する。

【0002】

【従来の技術】 従来、データベースの整合性をチェックする技術として、例えば、特開平 5-81110 号公報に記載のものが知られている。これは、インデックスファイル中のデータを物理的に検索したうえでインデックスキーでソートした結果を正解とし、一方、インデックスファイル中のデータをインデックスキーによりソートした結果を得て、これら両結果を比較することにより、インデックスファイルの整合性を検証するものである。

【0003】

【発明が解決しようとする課題】 特開平 5-81110

号公報に記載の技術は、データベースの整合性を検証するためのものであるが、これはデータベースのインデックス部には問題がないことを前提として検証を行っている。そのため、インデックス部に破壊や欠損などの問題が生じている場合には、インデックスをキーとしたチェック対象ページの検索ができないため、データベース中の問題箇所を特定できない事態が生じる。結局、データベースにおいてインデックス部に問題がある場合には、データベースの整合性チェックを正しくできないという問題がある。

【0004】 本発明の目的は、インデックス自体に破壊や欠損などの問題が生じている場合であってもデータベースの整合性チェックを可能とし、すべての破壊箇所を特定・検出するデータベースのインデックス整合性チェック処理方法、装置およびプログラムを提供することにある。

【0005】

【課題を解決するための手段】 上記目的を達成するために、本発明は、インデックス内のページ間を結ぶポインタ情報を意識することなく、物理順にページを入力し、インデックスページ情報を作成する手段と、作成されたインデックス情報を最上段から最下段まで段毎に並び替えを行い、各段情報を作成する手段と、最終的に作成される最下段の段情報からポインタの整合性のチェックを行う手段とを備えることを特徴とする。

【0006】 また、TREE 構造のデータが格納されたデータベースと計算機とが接続されたシステムにおけるインデックス整合性チェックプログラムにおいて、前述のプログラムは、前述のデータベースに格納された TREE 構造のデータの格納情報を取得し、前述のデータの格納情報に基づいて、前述の TREE 構造の階層ごとにデータを並べ替え結果と、前述の TREE 構造の最下位の階層でのデータの並べ替え結果とを比較し、前述の比較結果に基づいて、前述のデータベースに格納された TREE 構造のデータのデータ構造の整合性のチェックを行う。

【0007】

【発明の実施の形態】 以下、本発明を実施する場合の一形態を図面を参照して具体的に説明する。

【0008】 図 1 は、本発明の実施の一形態の TREE 構造型インデックスを備えるデータベースシステムの構成図である。データベースシステム 1 は、データベースに格納されたデータ部 3 を参照するためのインデックスを保持するインデックス部 2、実際のデータを保持するデータ部 3 からなるデータベース 4 を備え、さらに本発明におけるデータベースのインデックス整合性チェック処理を行う整合性チェック処理部 5 と、そのチェック結果が出力されるチェック結果保持部 6 から構成される。

【0009】 尚、図 1 では図示しないが、本発明におけるデータベースのインデックス整合性チェック処理を行う

プログラムが稼動する計算機と複数のデータベースとがネットワークを介して接続されているシステムにおいて、本発明を適用しても良い。

【0010】図2はインデクス部2の物理構造21を表したものである。インデクス部2は複数のインデクスページから構成され、各インデクスページは当該ページが格納されている物理ページ番号であるページ#2a、TREE構造における最上段、中段および最下段の階層を示し、最下段を「0001」とした各段の段数であるページ種別2b、各ページ種別において最上段以外の各段毎の順序番号であるエントリ2cを備える。インデクス部の物理構造21は、各インデクスページがデータセットの先頭からページ#2aの順に格納されている。

【0011】図3はインデクス部の論理構造22を表したものである。インデクス部2は図示したように、TREE構造を構成する最上段ページ、2～n段目ページ、最下段ページが、論理的な階層構造をなしている。

(尚、ここでは論理的な構造の例としてTREE構造を示したが、これ以外の構造でもよい。図3でページの一例を図示したが、これ以外のものをページとしても良い。)

最上段ページは下段ページの各格納最大キー値2d及び下段ページ#2eの組からなるインデクスエントリを、中段ページは最上段ページと同様に、下段ページの各格納最大キー値2d及び下段ページ#2eの組からなるインデクスエントリを持つ。下段ページ#とは当該ページが管理する直下の下段ページの物理ページ番号である。最下段ページは、最下段ページ内の前後のページ番号2f、2gおよびデータ部3内のエントリの位置情報であるキー値2hおよびそのデータが格納されているページのページ番号であるページ#2iを持つ。中段ページは複数段で構成される場合もあるが、本実施の形態における図3の例では中段ページは1段のみとしている。また、図3の例では3段の階層構造を構成しているため、最上段ページのページ種別は「0003」となる。

【0012】図4はメモリ上に作成するインデクス情報の作成処理手順を示すフローチャートである。

【0013】これは、本発明の目的であるインデクス自体の状態に関わらず整合性チェックを行うため、インデクスページを物理順に入力し、チェックに必要なインデクス情報を作成する処理である。(尚、フローチャートにおいて、ページ番号等の入力の処理は、本発明にかかるインデクス整合性チェックを行うプログラムが、データベースに格納されているページ情報等の処理に必要な情報を取得する処理にしてもよい。また、データベースからインデクス整合性チェックプログラムへページ情報等必要な情報を送信し、前記情報を受信したインデクス整合性チェックプログラムが必要な処理を行うようにしてもよい。他の処理におけるデータ入力処理も同様である。)

図5は作成されるインデクス情報の一具体例である。インデクス情報50aは入力ページ毎に作成され、項目としてページ#51、ページ種別52、エントリ53、次ページ#54、前ページ#55を持つ。以下、インデクス情報作成処理の詳細を説明する。まず、インデクスページを物理順に入力し(以下、入力ページ#)(ステップ401)、終了判定(ステップ402)を行い、入力終了でなければ、ページ種別の判定を行う(ステップ403)。

10 【0014】最上段ページの場合には、ページ内のインデクスエントリ数分のインデクス情報を作成する。各インデクス情報のページ#には入力ページ#の下段ページ#を設定し(ステップ404)、エントリにはステップ404で設定した各ページ#に対応するエントリ通番を設定し(ステップ405)、ページ種別には最上段種別を設定する(ステップ406)。

図3の例では、最初に入力される入力ページ#1は最上段ページであるから、各インデクス情報のページ#51には下段ページである「#8」と「#5」を設定する。そして、ページ#51に「#8」「#5」が設定された各インデクス情報に対応して、エントリ53にはそれぞれ「0001」「0002」を設定する。ページ種別52には最上段種別「0003」を設定する。それを示したのが、インデクス情報501、502である。

20 【0015】次に、入力ページ#が最上段ページでない場合、最下段ページであるかの判定(ステップ407)を行う。最下段ページでない場合には、入力ページ#のページ内にあるインデクスエントリ数分のインデクス情報を作成する。各インデクス情報のページ#には入力ページ#のページ#を設定し(ステップ408)、エントリには入力ページ内のエントリ通番を設定し(ステップ409)、ページ種別には中段種別を設定し(ステップ410)、次ページ#には各エントリに対応する下段ページ#を設定する(ステップ411)。

30 【0016】入力ページが図3の「#5」の場合、最上段ページでも最下段ページでもないため、下段ページ「#4」「#3」「#9」の3つインデクス情報を作成する。各インデクス情報のページ#51には自ページのページ#「#5」を設定し、エントリ53には順にエントリ「0001」「0002」「0003」を設定する。そしてページ種別52には中段種別「0002」を、次ページ#54には各下段ページ#2eを設定する。それを示したのが、インデクス情報503、504、505である。

40 【0017】ステップ407で最下段ページと判定した場合は、インデクス情報のページ#には入力ページ#を設定し(ステップ412)、種別には最下段種別を設定し(ステップ413)、次ページ#、前ページ#には、入力ページ#の横ポインタ情報である後ページ#2g、前ページ#2fを設定する(ステップ414)。

【0018】入力ページが図3の「#6」の場合、これは最下段ページであるから、インデクス情報のページ#51には入力ページ#の「#6」を設定し、ページ種別52には最下段種別「0001」を設定する。そして次ページ#54及び前ページ#55には、それぞれ自らの後ページ2g、前ページ2fである「#4」「#7」を設定する。それを示したのがインデクス情報506である。

【0019】すべてのページに対するインデクス情報作成処理が終了した後、作成したインデクス情報をページ種別（降順）、ページ#（昇順）、エントリ（昇順）で並び替え（ステップ415）、処理を終了する。この並び替えの結果、インデクス情報50bようになる。

【0020】次に、インデクス情報50bのページ種別52毎に、各段毎の情報作成処理を順次行う。

【0021】これは、データ構造の整合性チェックを行うために、物理順に入力し作成されたインデクス情報をインデクスの論理的構造を意識した順に並び替える処理である。図6は最上段情報作成の処理手順を示すフローチャートである。図7は作成された最上段情報70の具体例である。最上段情報70はページ#71とエントリ（1）72を有する。以下、先ず最上段情報の作成処理手順を詳細に説明する。初めに、インデクス情報50bから1件ずつインデクス情報を入力し（ステップ601）、ページ種別の判定を行い（ステップ602）、最上段ページでない場合、最上段情報作成処理を終了する。ステップ602において最上段ページと判定した場合、インデクス情報のページ#を最上段情報70のページ#71に設定し（ステップ603）、インデクス情報のエントリを最上段情報のエントリ（1）72に設定する（ステップ604）。図5のインデクス情報50bでは、最上段ページのページ種別「0003」に該当するのは507で示したインデクス情報であるから、これらがステップ603と604の処理対象となる。各インデクス情報のページ#51がインデクス情報のページ#71に、エントリ53がエントリ（1）72にそれぞれ設定される。その結果、図7に示されたような最上段情報70が作成される。

【0022】次に、図8は中段情報の作成処理手順を示すフローチャートである。図8において、まず処理対象情報の初期値として最上段情報を設定し（ステップ801）、処理番号iの初期値として1を設定し（ステップ802）、処理中ページ種別の初期値として2段目のページ種別を設定する（ステップ803）。次に、インデクス情報50bを入力し（ステップ804）、ページ種別を判定し（ステップ805）、それが処理中ページ種別と一致した場合は、処理対象情報からそのページ#と同じページ#を持つインデクス情報をサーチする（ステップ806）。

【0023】この時点では、処理対象情報は最上段情報

となっているので、入力する処理対象情報は、図7で示した最上段情報70である。また処理中ページ種別は初期値の「0002」であり、インデクス情報50bのインデクス情報のうち、ページ種別52が「0002」に該当するのは508で示したものである。これらが以下のステップ807～809の対象となる。n段目の中段情報は、ページ#、エントリ（1）、エントリ（2）、・・・、エントリ（n）で構成される。図9は2段目の中段情報の具体例であるが、これをもとにその作成処理の詳細を説明する。

【0024】まず、(i+1)段の中段情報のページ#91にはインデクス情報50bの次ページ#54を設定し（ステップ807）、1～i番目の各エントリには処理対象情報で同じページ#を持つインデクス情報の1～i番目のエントリを設定し（ステップ808）、(i+1)番目のエントリにはインデクス情報50bのエントリを設定する（ステップ809）。図5のインデクス情報508の場合、ページ#91には各インデクス情報の次ページ#54、エントリ（1）92には処理対象情報である最上段情報70のエントリ（1）72、エントリ（2）93にはエントリ53がそれぞれ設定される。それを示したのが中段情報90aである。

【0025】次に、入力したインデクス情報のページ種別が処理中ページ種別と一致しない場合、ページ種別を判定し（ステップ810）、最下段ページでない場合には処理対象情報として直前に作成した中段情報(i+1)を設定する（ステップ811）。この中段情報(i+1)をページ#91で昇順に並び替えを行い（ステップ812）、処理番号iを1カウントアップし（ステップ813）、処理中ページ種別に入力したインデクス情報のページ種別を設定し（ステップ814）、インデクス情報入力（ステップ804）へ戻る。ただし、ページ種別切り替え時には次のインデクス情報入力を行わず、直前に入力済みのインデクス情報を最初の処理対象とする。

【0026】具体的には、中段情報90aを作成後、次に入力したインデクス情報が最下段ページでない場合には、処理対象情報として中段情報90aを設定し、それをページ#91で昇順に並び替えを行い、処理番号iを1だけカウントアップしてi=2とする。そして処理中ページ種別に、入力したインデクス情報のページ種別52を設定して、ステップ804へ戻る。そして、この入力したインデクス情報から次の処理を行う。しかし本実施例においては、中段情報90a作成後に入力されるインデクス情報509は最下段ページであるため、本処理は行われない。

【0027】ステップ810で最下段ページと判定した場合には、直前に作成した中段情報（最下段より一段上の中段情報）をページ#で昇順に並び替え（ステップ815）、処理を終了する。その結果を示したのが中段情報

報90bである。

【0028】次に、最下段情報作成処理について説明する。図10は最下段情報作成の処理手順を示すフローチャートである。まず、処理対象情報として最終の中段情報（最下段より一段上の中段情報）を設定し（ステップ1001）、インデクス情報50bを入力し（ステップ1002）、終了判定（ステップ1003）を行う。具体例では、処理対象情報は中段情報90b、対象となるインデクス情報は図5の509である。ステップ1003で入力終了でなければ、入力したインデクス情報のページ#と同じページ#を持つインデクス情報を処理対象情報からサーチし（ステップ1004）、最下段情報の各エントリには処理対象情報の対応するすべてのエントリを設定し（ステップ1005）、ページ#113には処理対象情報のページ#を設定し（ステップ1006）、最下段情報の次ページ#113、前ページ#114にはインデクス情報の次ページ#54、前ページ#55を設定する（ステップ1007）。入力終了の場合、作成した最下段情報を各エントリの昇順で並び替えを行い（ステップ1008）、処理を終了する。

【0029】処理対象情報が中段情報90b、インデクス情報が図5で示した509である場合、最下段情報のエントリ（1）111、エントリ（2）には、処理対象情報90bのエントリ（1）92、エントリ（2）93を設定する。さらに、ページ#113にはページ#91、次ページ#114、前ページ#115にはそれぞれ対応する次ページ#54、前ページ#55を設定する。その結果が図11の最下段情報110aであり、それを各エントリの昇順で並び替えたものが図11の最下段情報110bである。

【0030】この図11の最下段情報をもとにデータベースのインデクス整合性チェックを行う。図11の最下段情報をもとにした整合性チェックの処理手順を示す図12のフローチャートを参照して、その処理を説明する。

【0031】まず、先頭の最下段情報を入力し（ステップ1201）、ページ#及び次ページ#をワーク領域に格納し（ステップ1202）、前ページ#の判定を行う（ステップ1203）。ここで前ページ#が「0000」でない、すなわち前ページ#が存在する場合はエラー情報の出力を行う（ステップ1204）。

【0032】最下段情報110bの例では、先頭のインデクス情報1101が入力されると、ページ#113の#2がワーク領域fに、次ページ#114の#7がワーク領域fにそれぞれ格納される。そして前ページ#115は「0000」であるから、次のステップへ進む。

【0033】更に、次の最下段情報を入力し（ステップ1205）、終了判定（ステップ1206）を行う。入力終了でなければ、入力したインデクス情報のページ#とワーク領域に格納した次ページ#および入力したイン

デクス情報の次ページ#とワーク領域に格納したページ#を判定し（ステップ1207）、一致しない場合はエラー情報を出力し（ステップ1208）、ページ#及び次ページ#をワーク領域に格納し（ステップ1209）、最下段情報の入力（ステップ1205）に戻る。

【0034】インデクス情報1102が入力されると、そのページ#113である#7とワーク領域fに格納されているページ#を比較判定する。ワーク領域fのページ#は#7であり、一致していると判定するので、インデクス情報1102のページ#の#7をワーク領域fへ、次ページ#の#6をワーク領域fへ格納し、ステップ1205へ戻る。

【0035】入力終了の場合、最後に入力されたインデクス情報の次ページ#の判定を行い（ステップ1210）、次ページ#が「0000」でない場合、すなわち次ページが存在すると判定した場合にはエラー情報の出力を行い（ステップ1211）、「0000」である場合には本インデクス整合性チェック処理を終了する。

【0036】図13はインデクスに不整合が生じている場合の一具体例である。ここでページ#5は下段ページ#としてページ#3にリンクされているはずが、誤って存在しないページ#である#10にリンクされている。また、最下段ページ#7の後ページは、ページ#6となるはずが後ページがない状態になっている。

【0037】このような不整合を抱えたインデクスに対して、本実施の形態で述べた一連の処理を行った結果が図14に示したものであり、最下段情報110bのうち最下段情報チェック処理の結果が、エラー情報出力140のように示される。これらエラー情報出力はチェック結果保持部6にチェック結果として格納される。エラー情報140aはページ#7に次ページ#がないことを示している。エラー情報140b、140cはそれ自体にエラー原因があるわけではないが、両出力のエントリ

（2）が連続していない。つまり0002と言う値を持つエントリ（1）のエントリ（2）の値0002が欠損しており、ページ#4とページ#9との間でページが一つ抜けていることを示している。ここで、2段目ページのエントリ（1）のうち0002はページ#5を示し、かつそれに属するエントリ（2）のうち0002はページ#10である。よってこのページ#5とページ#10とのリンクがエラーであり、正しいページは140bの次ページ#及び140cの前ページ#よりページ#3であることが分かる。

【0038】以上、説明したように、本発明によれば、TREE構造型インデクスページを入力し、各ページのインデクス情報を作成すると共に、そのインデクス情報から各段の情報を作成し、最終的に最下段情報を作成することで、容易にインデクスページの整合性チェックができる。

【0039】また、各インデクスページの入力は、物理

10

20

30

40

50

順に一回だけで済み、入力回数は最小限に抑えられることにより、チェック性能が向上する。

【0040】また、入力時にはインデクス内のポインタ情報を意識しないため、途中で処理が続行不可になることなく、複数の破壊個所を1回でチェック可能になる。

【0041】また、本発明を適用したデータ構造の整合性チェックを行うプログラムによって、データベースに格納されたTREE構造のデータの格納情報を取得し、取得したデータの格納情報に基づいて、TREE構造の階層ごとにデータを並べ替えた結果とTREE構造の最下位の階層でのデータの並べ替え結果とを比較し（もしくは、TREE構造の縦方向に繋がったデータ構造と、TREE構造の同じ階層ごと（同じ段ごと）に繋がったデータ構造とを比較し）、比較結果に基づいてデータベースに格納されたTREE構造のデータのデータ構造の整合性のチェックを行うことが可能になる。

【0042】本発明を適用したプログラムにより、データ構造の破損個所をチェックするで、データ構造の修復する際の処理を軽減することが可能になる。

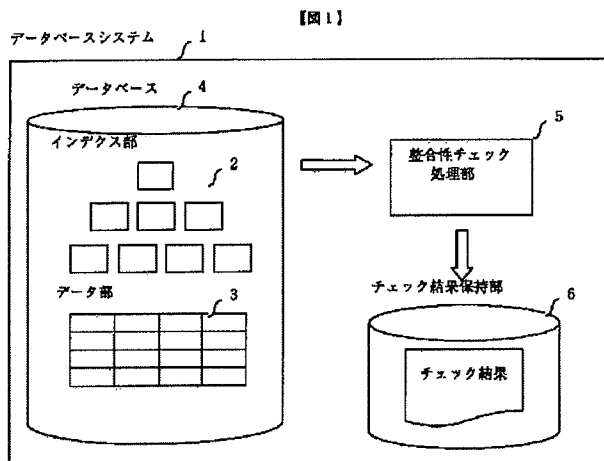
【0043】

【発明の効果】以上、説明したように、本発明によれば、TREE構造型インデクスページを入力し、各ページのインデクス情報を作成すると共に、そのインデクス情報から各段の情報を作成し、最終的に最下段情報を作成することで、インデクスページの整合性チェックができる。

【図面の簡単な説明】

【図1】本発明の一実施形態のデータベースシステムの

【図1】



概略構成を示すブロック図、

【図2】インデクス部の物理構造を示す図、

【図3】インデクス部の論理構造を示す図、

【図4】インデクス情報作成処理の手順の概要を示すフローチャート、

【図5】インデクス情報の概要構成を説明するための図、

【図6】最上段情報作成処理の手順の概要を示すフローチャート、

10 【図7】最上段情報の概要構成を説明するための図、

【図8】中段情報作成処理の手順の概要を示すフローチャート、

【図9】中段情報の概要構成を説明するための図、

【図10】最下段情報作成処理の手順の概要を示すフローチャート、

【図11】最下段情報の概要構成を説明するための図、

【図12】最下段情報チェック処理の手順の概要を示すフローチャート、

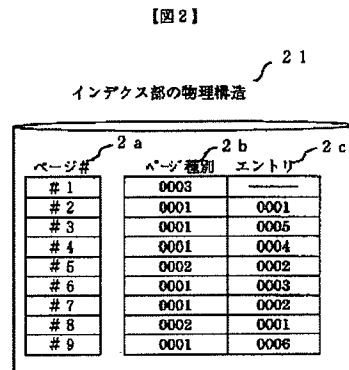
20 【図13】インデクス部に不整合がある論理構造の一具体例、

【図14】インデクス部に不整合がある場合のチェック処理の結果を示す図、

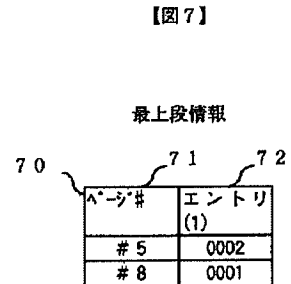
【符号の説明】

1・・・データベースシステム、2・・・インデクス部、3・・・データ部、4・・・データベース、5・・・整合性チェック処理部、6・・・チェック結果保持部、21・・・インデクス部の物理構造、22・・・インデクスの論理構造。

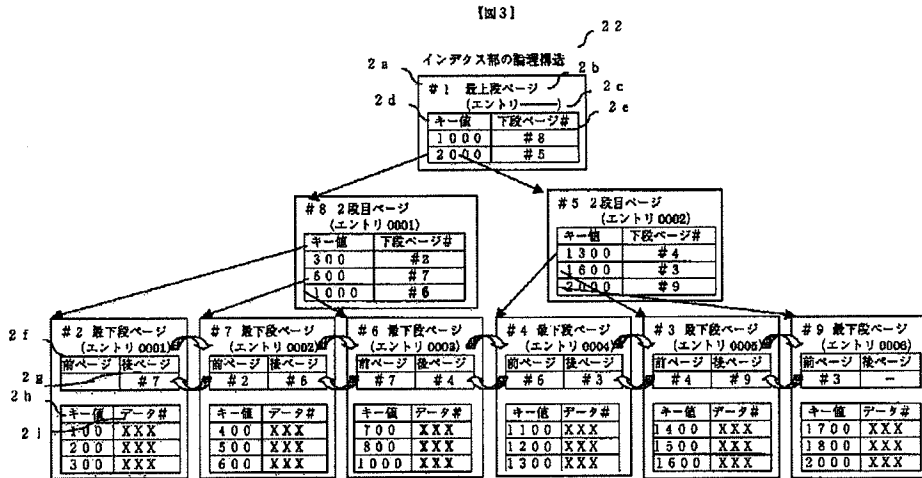
【図2】



【図7】

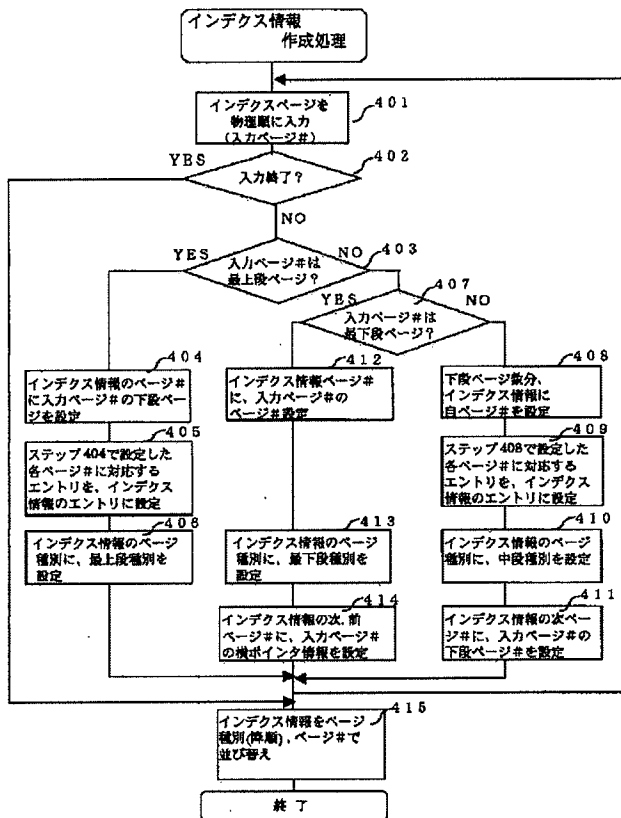


【図3】



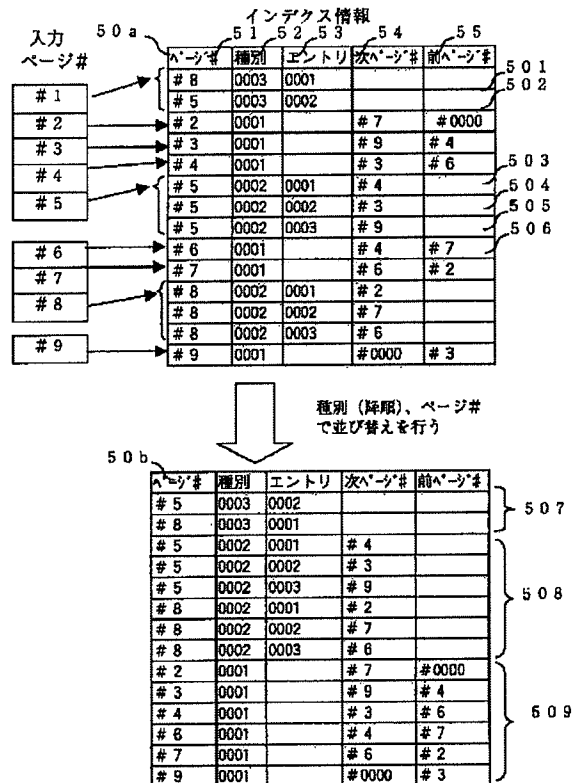
【図4】

【図4】



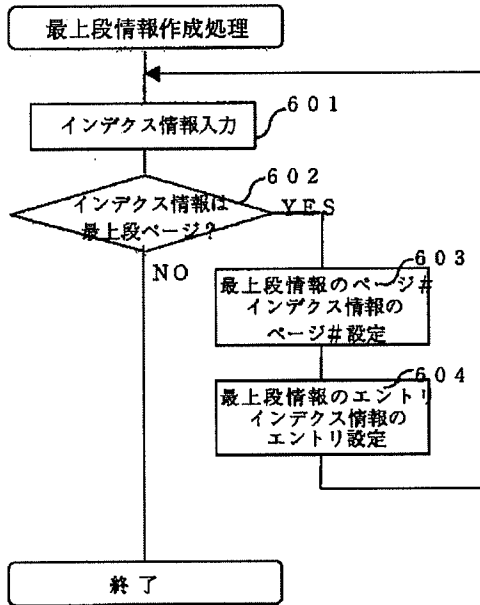
【図5】

【図5】



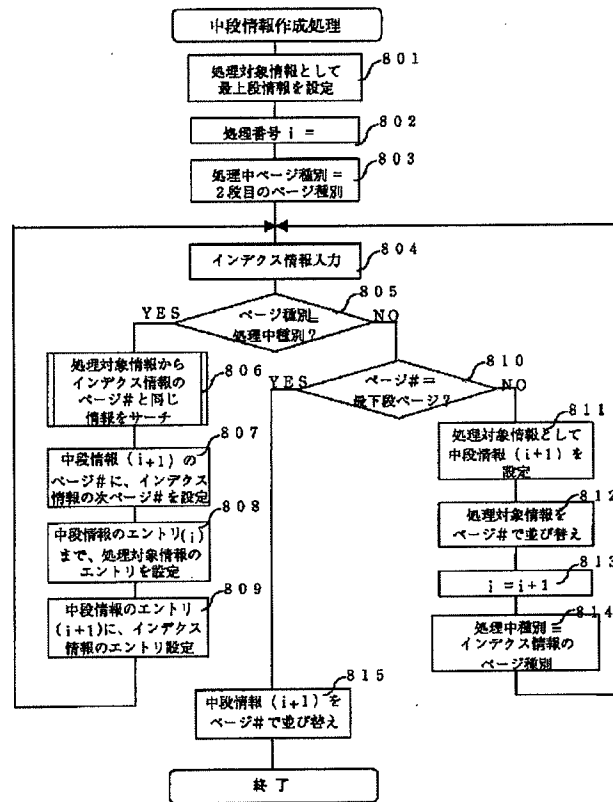
【図6】

【図6】



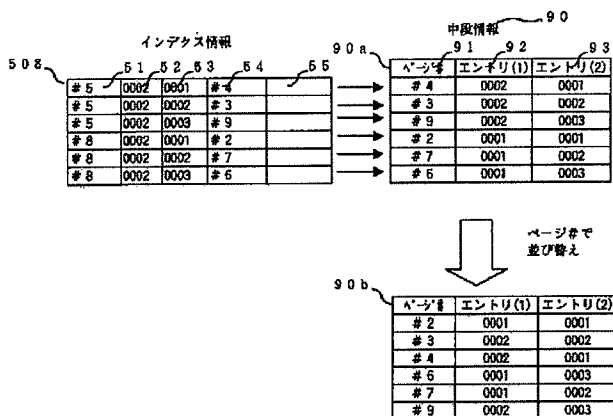
【図8】

【図8】



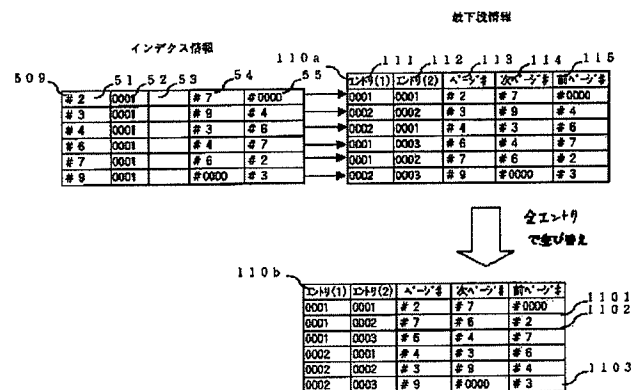
【図9】

【図9】



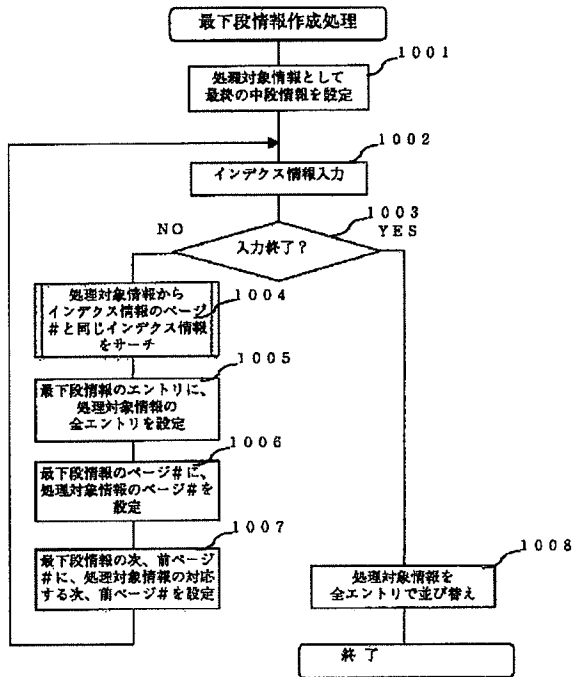
【図11】

【図11】



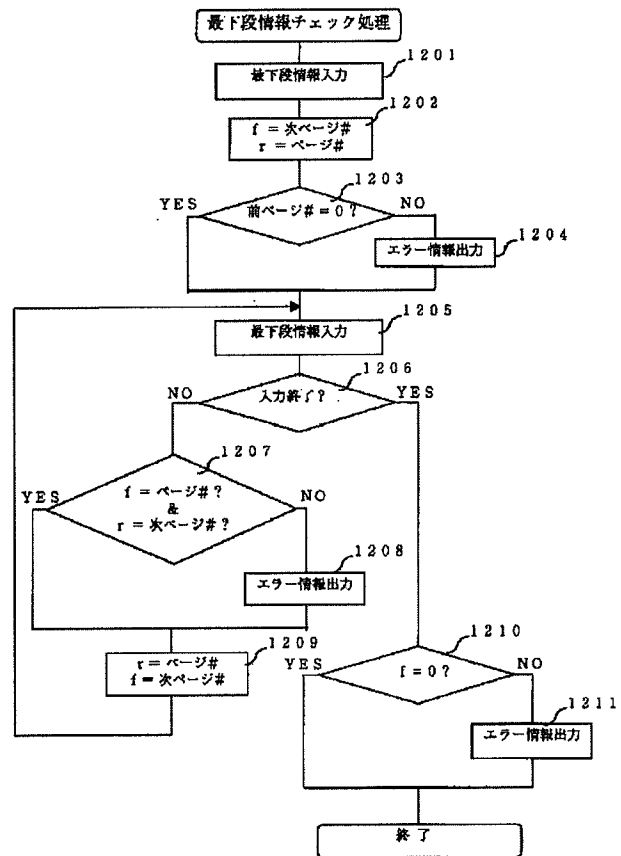
【図10】

【図10】



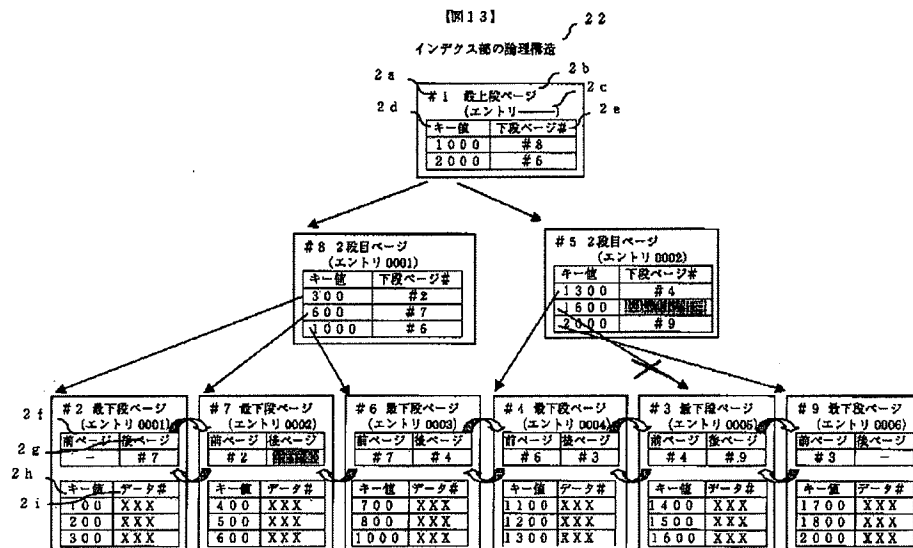
【図12】

【図12】



【図13】

【図13】



【図14】

【図14】

